



## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **10143489 A**(43) Date of publication of application: **29 . 05 . 98**

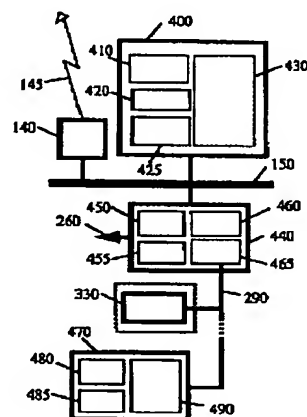
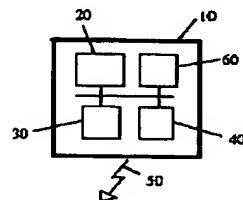
(51) Int. Cl

**G06F 15/78**(21) Application number: **08298420**(22) Date of filing: **11 . 11 . 96**(71) Applicant: **HITACHI LTD**(72) Inventor: **TONOMURA MOTONOBU  
ITO KIYOO****(54) INFORMATION PROCESSING SYSTEM****(57) Abstract:**

**PROBLEM TO BE SOLVED:** To efficiently perform a processing as the entire system by leaving the management of a built-in memory loaded by a function memory chip to the function memory itself.

**SOLUTION:** As a function to be co-mounted to a memory such as a DRAM or the like, not only arithmetic/control logic but also a central processing unit (CPU) function 480 for controlling them by programming 485 is provided and turned to a function memory chip 470. Thus, a control program dedicated to a function memory chip 470 on the side of a multi-media processor(MMP) is shifted to a control memory (CM) 485 on the side of the function memory chip 470 and the control program on the side of the MMP is reduced to the issuance of only the instruction of a function to be requested to the function memory chip 470. Since the MMP is loaded with the control memory, it can be considered as the function memory chip. Also, since the DRAM can be co-mounted, it is turned to the function memory chip 440.

COPYRIGHT: (C)1998,JPO



(19)日本国特許庁 (J P)

## (12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-143489

(43)公開日 平成10年(1998)5月29日

(51)Int.Cl.<sup>8</sup>

G 0 6 F 15/78

識別記号

5 1 0

F I

G 0 6 F 15/78

5 1 0 C

審査請求 未請求 請求項の数9 O L (全 8 頁)

(21)出願番号 特願平8-298420

(22)出願日 平成8年(1996)11月11日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 外村 元伸

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72)発明者 伊藤 清男

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(74)代理人 弁理士 小川 勝男

(54)【発明の名称】 情報処理システム

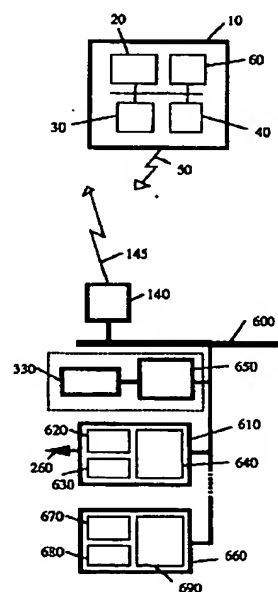
## (57)【要約】

【課題】 MPUとDRAMによって構成される従来システムには、転送ネックによってシステム性能に自ずと限界があり、MPUあるいは周辺チップ側に様々な機能が追加搭載されると、それぞれに対応されるため、チップ間のインタフェースが異なり、汎用性が低くなり、拡張の容易性や低価格化が期待できない。

【解決手段】 DRAM部とそのアクセス制御と処理をするCPU部を混載して様々な機能をもつ機能メモリ・チップを設けて相互に接続し、コマンド・データ転送により機能メモリ・チップ間アクセスを統一、簡略、高速かつ円滑化することで達成される。

【効果】 情報処理システムでは、転送速度ネックとなる機能が、同一チップ内で実現され、外部チップへのデータ転送回数が低減でき、システム全体の性能を飛躍的に向上させることができる。また、拡張性が容易かつ安価なシステムを構築できるという効果がある。

図 6



## 【特許請求の範囲】

【請求項1】論理演算または中央演算処理（CPU）機能をもつ部とメモリ部からなり、1本または2本以上の入出力線を設け、該入出力線を用いて転送されるコマンド・データのエンコードおよびデコード部を設け、該入出力線の入力線と出力線の数制御することを特徴とする機能メモリ・チップ。

【請求項2】特許請求項1に記載の複数の機能メモリ・チップは、入出力線で相互に接続され、該機能メモリ・チップの1つが複数の他の機能メモリ・チップに対して同時に転送先機能メモリ・チップ名をコマンド転送し、それぞれの機能メモリ・チップで比較照合して一致した機能メモリ・チップをアクセスすることを特徴とする情報処理システム。

【請求項3】アクセスされる機能メモリ・チップ内処理時間の長短に応じて入出力線の占有権を制御することを特徴とする請求項2に記載の情報処理システム。

【請求項4】機能メモリ・チップ間のアクセス頻度に応じて入出力線の入力線と出力線の数制御することを特徴とする請求項2に記載の情報処理システム。

【請求項5】各機能メモリ・チップの命令コード体系は、それぞれ独立であることを特徴とする請求項2に記載の情報処理システム。

【請求項6】異なる機能に特化したことを特徴とする請求項2に記載の情報処理システム。

【請求項7】少なくとも2種類の機能メモリ・チップの一方に出力結果を出す場合に、該機能メモリ・チップの処理実行時間が他の機能メモリ・チップの処理実行時間と該機能メモリ・チップ間転送時間の和より小さくなるように該機能メモリ・チップをプログラミングすることを特徴とする特許請求項2に記載の情報処理システム。

【請求項8】メモリ部はDRAMを含むことを特徴とする特許請求項1に記載の機能メモリ・チップ。

【請求項9】論理演算またはCPU機能をもつ部とメモリ部からなり、1本または2本以上の入出力線を設け、該入出力線を用いて転送されるコマンド・データのエンコードおよびデコード部を設け、該入出力線の入力線と出力線の数制御する機能メモリ・チップ、マイクロプロセッサ・ユニットおよびメモリ・コントローラを持つことを特徴とする情報処理システム。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、中央演算処理（CPU）機能などを埋め込んだメモリ・チップとそれらによって構成されるシステムに係わり、特にDRAMのような大規模半導体メモリ・チップ内にCPU機能を埋め込んでチップ内で効率的に特定の処理機能をプログラミングして解釈実行するのに好適な情報処理システムに関する。

## 【0002】

【従来の技術】安価なネットワーク端末を構成する例として、図1に示すように、従来の汎用マイクロプロセッサ・ユニット（MPU）100に大容量DRAM130とそのメモリ・コントローラ（MC）120を搭載し、周辺コントローラ（PEC）160と外部メモリ（フラッシュ・メモリ170やROM180など）をコントロールするMC165を搭載し、ネットワーク・コントローラ（NC）140によってネットワーク回線145、50でサーバ10と結び、MPU100、PEC160とNC140はバス150によって相互に接続されているシステムなどが考えられている（日経マイクロデバイス、1996.03, pp. 60-63）。PEC160は、前記バス150とは別に映像機器に出力するAV信号168がある。サーバ10は、MPU20、主メモリ30、2次メモリ60およびNC40などで構成されている。

【0003】また、最近のゲーム機器用チップセットなどを利用して安価なネットワーク端末を構成する例として、図2に示すように、中央演算処理装置（CPU）210、キャッシュ・メモリ225、メモリ管理ユニット（MMU）220からなる汎用マイクロプロセッサ・ユニット（MPU）200、画像・グラフィック専用処理およびそのコントロール・メモリ（CM）250とDRAM270、280などを制御するメモリ・コントローラ（MC）255を内蔵したマルチメディア・プロセッサ（MMP）230、およびネットワーク回線145でサーバ10と結ぶネットワーク・コントローラ（NC）140からなり、MPU200、MMP230とNC140はバス150によって相互に接続されているシステムが考えられる（日経エレクトロニクス、1996.01.15, pp. 16-17）。MMP230には、前記バス150とは別に映像機器に出力するAV信号260がある。

## 【0004】

【発明が解決しようとする課題】上記従来例では、DRAMなどのメモリ単独チップに格納されている内容は、どうしてもCPU機能をもったチップに一旦ローディングして処理し、結果をまた戻してストアしなければならないため、システム性能には自ずと限界がある。高速転送機能を追加しても、ランダム・アクセス性能の向上はあまり期待できない。ランダム・アクセスはメモリ・チップ側で処理の方が効率的である。

【0005】また、高性能なマルチメディア処理機能その他多くの機能をMPUあるいは周辺チップ側に追加搭載することが要求されているため、いくつかの組み合わせ的方法により、各チップの様々な機能拡張が行われるが、それぞれに対応されるため、チップ間のインタフェース（ピン数、配置）が異なることが予想され、汎用性が低くなり、システム構成の開発コストの増加により低価格化が期待できない。様々な機能をもったチップのインタフェースを統一するあるいは吸収するしくみにより、汎用性を高め、従来のメモリ・チップのような低価格性と拡張性のよいものを提供する必要がある。

## 【0006】

【課題を解決するための手段】本発明は、メモリ部と中央演算処理（CPU）部などからなり、チップ・インタフェースとして少なくとも入出力線をもつ機能メモリ・チップを設け、入出力線をコマンド・データ転送路として使用し、コマンド・データのデコード／エンコードにより、指定機能メモリ・チップ名と比較照合して機能メモリ・チップ内のメモリ部に自由にアクセスし、処理ネックとなる機能はその機能を最適実行できる構成をもった機能メモリ・チップで実現し、アクセス・ネックとなる機能は一つの機能メモリ・チップ内でまとめて最適プログラミング実現し、いくつかの異なる機能メモリ・チップを入出力線で相互に接続して構成し、システム全体として効率的に処理できる情報処理システムを提供することにある。

【0007】

【発明の実施の形態】最近では、高性能な3次元グラフィックス、アニメーションや映像の圧縮／伸長機能が必須になりつつある。そこで、画像メモリとしてDRAMなどが使われている従来のシステム構成に加えて、図3に示すように、画像メモリとその演算・制御論理を混載させた高性能マルチメディア・チップ（ALM）300をメモリ・バス290に拡張接続することが考えられる。こうすれば、従来のマルチメディア・プロセッサ（MMP）230に容易に接続して、機能拡張を実現できる。欠点としては、ALM300の専用コントロール・プログラムをMMP230側のコントロール・メモリ（CM）250に載せなければならないので、仕様の異なるALMを接続するたびにMMP-ALM専用のプログラムをゼロから開発しなければならないことがある。

【0008】このような欠点を克服する手段として、DRAMなどのメモリに混載させる機能として演算・制御論理ばかりでなく、図4に示すように、これらをプログラミング485によってコントロールする中央演算処理（CPU）機能480を設け、機能メモリ・チップ470とする。この手段によって、MMP230側の機能メモリ・チップ470専用コントロール・プログラムは、機能メモリ・チップ470側のコントロール・メモリ（CM）485に移り、MMP230側のコントロール・プログラムは機能メモリ・チップ470に依頼する機能の命令のみの発行に軽減される。

【0009】今後の拡張性を考えた場合、チップ・インタフェースだけを統一しておけば、新規開発される機能メモリ・チップはどんな専用機能をもっている、またシステム共通の中間コードを媒介とすれば、どんな命令セットをもったCPUでも、どんな種類のメモリを混載していてもかまわない。MMP230はコントロール・メモリ250を搭載しているので、一種の機能メモリ・チップと考えてもよく、またDRAMを混載してもよいので、これを機能メモリ・チップ440とする。主メモリのDRAM330はなくてもよい。また、MPU400側にDR

AM430を搭載してもよい。したがって、MPU400自体も一種の機能メモリ・チップとする考え方も成り立つ。ただし、MPU400にはこのシステム全体のメモリ空間を管理するメモリ管理ユニット（MMU）420があるので、アドレス用ピン数は全メモリ空間相当をサポートする分を用意しなければならない。

【0010】もし、図5に示すように、ネットワーク端末などに必要なすべての機能を1チップ500で実現できれば、2チップ以上の構成にしていたときに必要だった機能や制約がなくせる。すなわち、メモリ管理がMPUチップ500内メモリ空間のものに簡略でき、従来のアドレス／データ・ピンはネットワーク用の入出力線145のピンのみになる。欠点としては、全機能を1チップに集約するために、当然ハードウェア量が増加しチップ面積が増大することである。その結果パッケージングも高価なものになる。内蔵メモリ560の容量が限定されるので、コンパクトな命令セット・アーキテクチャやプログラミングが必要になる。

【0011】MPUとメモリ単独チップ群によって構成される従来システムの主な課題は、MPU側が全メモリ空間をサポート管理しなければならないことと、MPUがメモリ・コントローラ（MC）によってどのメモリ・チップにアクセスするかを選択しなければならないことである。このことは、中央演算処理（CPU）機能とDRAMなどの主メモリを混載させる場合に、CPUとメモリのどちらを主体とさせるかという問題を生じさせる。すなわち、MPU400側に主メモリ430を搭載する場合は、内蔵および外付けのメモリを管理するメモリ管理ユニット（MMU）420をMPU400内に設けなければならない。メモリを主体としてメモリ側にCPU機能480を埋め込む機能メモリ・チップ470の場合は、どこかにメモリ・コントローラ（MC）465を設けなければならない。また、機能メモリ・チップ470を接続すると想定したときのMPU400のメモリ管理は、MPU400から依頼して機能メモリ・チップ470が動作している間に、機能メモリ・チップ470がアクセスするアドレス・エリア490をMPU400側でどのように管理するかを考えなければならない。

【0012】このような問題を単純にするための考え方の一つに、機能メモリ・チップ470が搭載している内蔵メモリ485、490の管理はその機能メモリ自身にまかせることにし、MPU400側では行わないとする方法がある。また、MPU／機能メモリ・チップがアクセスする機能メモリ・チップの選択は機能メモリ・チップ内の特定エリアに登録されている名前の照合によって行うことにし、MC465を必要としない方法がある。そうすることによって、図6に示すように、従来タイプのMPU400とMC465、650がなくても、機能メモリ・チップ610、660だけでシステムを構成することが可能になり、システム拡張が容易な機能分散型のメモリ・チップを安価に提

供することができる。

【0013】図6に示す情報処理システムはホストCPUとなるMPUを持たなくても、自立的にシステムを構築できるが、各機能メモリ・チップ610、660の性能を最大限に引き出す高性能なシステムを構築するためには、図7に示すように、機能メモリ・チップの一つ700が残りの機能メモリ・チップ群610、660のプロセス実行720をオブジェクト指向的にスケジューリング管理する役目710をもっていることが望ましい。オブジェクト・メモリ720には、オブジェクトと呼ばれる一固まりの機能（プロセス）を実行するプログラムが書かれており、そのプログラムの解釈はコントロール・メモリ部760で展開されて実行される。一固まりの機能には、オブジェクト名が付けられ、オブジェクト・メモリ720へのポイントとともにテーブル710にて管理される。また、この機能メモリ・チップには、オブジェクト・メモリ720の空きエリアを回収するガーベジコレクション（GC）機能770が備わっていることが望ましい。

【0014】従来システムと本発明の情報処理システムの違いについて図8と図9を用いて詳細に説明する。図8（a）の従来構成に示すように、従来システムのMPU400は、 $n$ 本のアドレス線と $j$ 本のデータ線をもっているとする。また、DRAMチップ330は、 $m$ 本のアドレス線と $i$ 本のデータ線をもっているとする。アドレス線292とデータ線291がマルチプレックスされているかもしれないが、説明上別々に考える。一般に、 $n > m$ と $j > i$ の関係にあるので、MPU400とDRAMチップ330の間にメモリ・コントローラ（MC）650を設けて、メモリ・チップとワード構成の選択を行う。

【0015】図8（b）の構成に示すように、本発明の機能メモリ・チップ610、660は、従来のDRAMチップ330と同じ $m$ 本のアドレス線と $i$ 本のデータ線をもっているとする。各機能メモリ・チップ610、660は、CPU部620、670とメモリ部630、640、680、690からなり、メモリ部の一部はコントロール・メモリ（CM）部630、680になっている。メモリ部は1バンク $i$ ビットからなる複数個（ $j/i$ 個）のバンクで構成されているとする。機能メモリ・チップ610、660のこのような構成によって、メモリ・コントローラ（MC）650のもとでは、機能メモリ・チップ内のDRAM部は、従来のDRAM単独チップと同じインタフェースを用いてアクセスできる。すなわち、コントロール・メモリ部630、680において、DRAM単独チップと同じ働きをするようにエミュレーションするプログラムを作成しコマンド命令の発行によりエミュレーション実行させる。

【0016】次に、MC650を設けなくて各機能メモリ・チップ610、660間のデータ・アクセスを可能にするしくみについて説明する。今度は、アドレス線292とデータ線291の区別をしないで、合計 $k = m + i$ 本の入出力線291、292のみをもっているものとする。これら入出力

線291、292を使って、アドレスやデータなどの情報を含めたコマンドのかたちにして指定の機能メモリ・チップに転送する方式をとる。

【0017】まず、機能メモリ・チップ610から機能メモリ・チップ660に処理を依頼する場合、他の機能メモリ・チップから出力依頼が出されていないことを出力要求線信号850によって確認後、機能メモリ・チップ610から出力要求線信号850にデータ出力要求を出す。それと同時に、機能メモリ・チップ610から機能メモリ・チップ660の名前データ800を入出力線291、292に出力する。各機能メモリ・チップは出力要求線信号850がオンになったことにより機能メモリ・チップ660の名前データ800を取り込み、図9に示すように、機能メモリ・チップ660のみが機能メモリ・チップ名が一致したとし、データ入力ゲート901を開いた状態のまま、データ入力要求を入力要求線851に出す。その他機能メモリ・チップはデータ入力ゲート901を閉じた状態にする。機能メモリ・チップ610は、入力要求線信号851のオンを確認したら、機能メモリ・チップ660に処理の依頼をする命令データをコマンド転送する。機能メモリ・チップ660側では、コマンド・データ810、820、830のプロトコルをデコードし、指定先頭アドレス822から指定された長さ分823の連続データ830を書き込む。機能メモリ・チップ660内へのデータの書き込みは、その機能メモリ・チップ内のバンク構成で最適にアクセスできるように機能メモリ・チップ610とは無関係に配置制御可能である。書き込みが終了すると、命令データ830の解釈が行われ、具体的な処理が実行される。

【0018】処理が実行され、結果が得られると、今度は、結果転送のコマンド・データが組み立てられ、機能メモリ・チップ660から機能メモリ・チップ610へ結果が転送される。結果転送の1つの手順は、機能メモリ・チップ610と機能メモリ・チップ660の役割を入れ替えるだけで前述の処理依頼転送と同じである。この手順では、機能メモリ・チップ610は機能メモリ・チップ660に処理を依頼したら、処理結果が出るまで別の機能メモリがデータ転送をできるように、データ送受権を放棄する。これは機能メモリ・チップ660での処理時間が長い場合に有効な方法である。

【0019】結果転送の別の手順は、機能メモリ・チップ610を結果待ちの状態にしておき、機能メモリ・チップ660が結果を出したらただちに転送してもらう方式である。転送先の機能メモリ・チップ610の名前を確認する必要がないので、処理がはやい。これは機能メモリ・チップ660での処理時間が短い場合に有効な方法である。結果転送のさらに別の手順は、機能メモリ・チップ610と機能メモリ・チップ660との間で頻繁に情報のやりとりをしなければならない場合に、 $k$ 本の入出力線291、292を送信線291と受信線292専用に分列分割する方法である。双方に割り当てる線の本数も、頻度に応じて重

み付けもできる。このように、入出力線291、292のみを使ってコマンド・データ転送する方式は、場合に応じて柔軟に対処できる転送処理方法を選択できるという利点がある。

【0020】次に、図8(c)のプロトコルを用いて、コマンド・データ転送のプロトコルについて説明する。コマンド・データはデータのリード/ライト/その他を示すコマンド部810、812とアドレス情報を含むデータ部820、830、800からなる。機能メモリ・チップ名800を指定するコマンド812がその他にはいる。リード/ライト・コマンド810は、機能メモリ・チップ内先頭アドレス822を指定するために、ビット/バイト/ワード/ロングワードなどのタイプ821でアドレスを表現し、データ長823を与える。そして、そのあとに連続データ830がつく。コマンド部810、812には、リード/ライト指定の他に、転送権利の継続/非継続状態を示すビットを設け、結果待ちのプロトコルの簡略化を可能にする。また、その他コマンドには、転送路のリード/ライト専用分割指定もできるものを設ける。

【0021】図10は、本発明の情報処理システム全体を効果的に構築するためのコンセプトを与える図である。すなわち、処理ネックとなる機能はその機能に特化した最適な専用の機能メモリ・チップ1000で実現し、アクセス・ネックとなる機能は一つの機能メモリ・チップ1010内でまとめて最適プログラミング実現できるようにし、外部チップ1011へのデータ転送回数を低減することにより、システム全体として効率的に処理できる組み合わせを選択する。アクセス頻度は、太さをもった矢印1020、1021、1022で表され、例えば太い矢印1021の方が細い矢印1020よりもアクセス頻度が多いことを示す。実現すると効果がありそうな専用の処理機能には、例えば、グラフィックス、画像の圧縮/伸長、音声/画像の認識、通信、辞書/データベース検索などがある。このような機能は、処理負荷が重いので専用のハードウェアを設けないと所望の性能を達成できないことが多い。

【0022】また、システム全体で考えて命令コード体系の中にこれら新機能の拡張命令を追加していたのでは、命令コード体系が複雑になり命令デコードの効率が悪くなる。そこで、各専用機能メモリ・チップ内ではその中で使用する命令のみに特化した独自の命令コードを設けることができるため、命令コード体系が単純になり、デコードの効率が向上する効果がある。言い換えれば、機能メモリ・チップ名が各機能を区別する一種の命令コードの役割を担っているわけで、機能メモリ・チップ内ではそれぞれ独自の命令コード体系を定義できるのである。

【0023】また、機能メモリ・チップ間にまたがるプログラミングをしていたのでは、両チップ間転送時間の影響が無視できなくなることがある。例えば、機能メモリ1010から機能メモリ1011に専用処理を依頼し、その結

果をまた機能メモリ1010で使って処理する場合、トータルの処理時間は、機能メモリ1011での処理実行時間に加えて機能メモリ間転送時間の和となるので、もし機能メモリ1011で処理する実行時間よりも機能メモリ・チップ1010内で多少の処理実行時間がかかってもトータル処理時間が少ないように代替できるならば、機能メモリ・チップ1010内で実行する方が効率がよいことになる。したがって、プログラミングするときは、このような最適化を考慮しなければならない。機能メモリ・チップ1010と1011間の転送を削減することは、他の機能メモリ・チップ1012、1013などのデータ転送に余裕ができることになり、システム全体としての性能向上効果も期待できる。

【0024】

【発明の効果】以上説明したような情報処理システムは、拡張性が容易かつ安価に構築できるばかりでなく、従来のMPU-メモリ・システムで問題となっていた転送速度の限界によるシステム性能の限界を飛躍的に向上させることができる。すなわち、転送速度ネックとなる機能は、同一チップ内に閉じこめて実現するような組み合わせにすれば、外部チップへのデータ転送回数が低減できるからである。

【図面の簡単な説明】

【図1】ネットワーク端末を構成する従来例1。

【図2】ネットワーク端末を構成する従来例2。

【図3】論理演算混載DRAMをメモリ・コントローラで接続したシステム。

【図4】メモリ・コントローラで接続した機能メモリ・チップ。

【図5】1チップ機能メモリ。

【図6】メモリ・コントローラによって接続する必要のない機能メモリ・チップ構成。

【図7】機能メモリ・チップ群のスケジューリング機能付き機能メモリ・チップ。

【図8】DRAMと機能メモリ・チップの接続法。

【図9】指定機能メモリ・チップ名を持つチップの選択機構。

【図10】機能メモリ・チップ群からなる情報処理システムのコンセプトを示す図。

【符号の説明】

10・・・サーバ、 20、100、200、400・・・マイクロプロセッサ・ユニット(MPU)、 30・・・主メモリ、 40、140、540・・・ネットワーク・コントローラ(NC)、 50、145・・・ネットワーク回線、 60・・・2次メモリ、 110、162、210、240、410、450、480、510、620、670、750・・・中央演算処理装置(CPU)、 120、165、255、465、650・・・メモリ・コントローラ(MC)、 125、225、425、530・・・キャッシュ・メモリ、 130、320、460、490、560、640、690・・・DRAM、 160・・・周辺コントローラ、 168、260・・・音声/映像信号、 170・・・フラッシュ

・メモリ、180・・・ROM、190、290・・・メモリ接続線、291・・・データ線、292・・・アドレス線、220、420、520・・・メモリ管理ユニット(MMU)、150・・・バス、270、280、330・・・DRAMチップ、230・・・マルチメディア・プロセッサ(MMP)、250、455、485、550、630、680、760・・・コントロール・メモリ(CM)、300・・・論理演算混載DRAMメモリ・チップ、310・・・論理部、440、610・・・機能メモリ・チップ1、470、660・・・機能メモリ・チップ2、500・・・1チップ機能メモリ、600・・・入出力線バス、700・・・スケジューリング機能メモリ・チップ、710・・・オブジェクト管理、720・・・オブジェクト・メモリ、730・・・オブジェクト名格納エリア、770・・・ガベージコレクション(GC)、850・・・出力要求線、851・・・入力要求線、800・・・機能メモリ・チップ

\* プ名部、810、812・・・コマンド部、820・・・アドレス指定部、821・・・ビット/バイト/ワード/ロングワードのタイプ指定部、822・・・先頭アドレス部、823・・・データ長部、830・・・データ部、900・・・プロトコル・デコーダ、901・・・入力、902、905・・・プロトコル・エンコーダ、906・・・出力、907・・・アドレス出力、910・・・機能メモリ・チップ名一致信号、925・・・アドレス入力、920・・・データ入力、921・・・データ出力、922・・・内部データバス、930・・・プログラム・カウンタ、641・・・機能メモリ・チップ名登録エリア、950・・・命令デコーダ、951・・・ハードウェア制御信号、960・・・算術論理演算器(ALU)、1000、1010、1011、1012、1013・・・機能メモリ・チップ、1020、1021、1022・・・線の太さがアクセス頻度の大小を表す。

【図1】

【図2】

【図3】

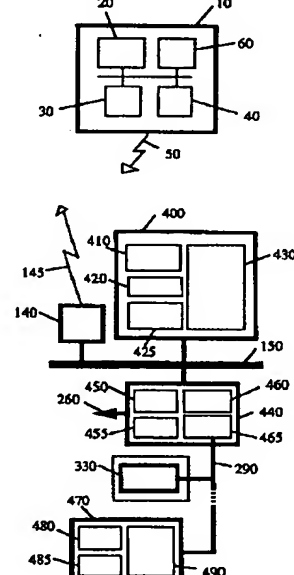
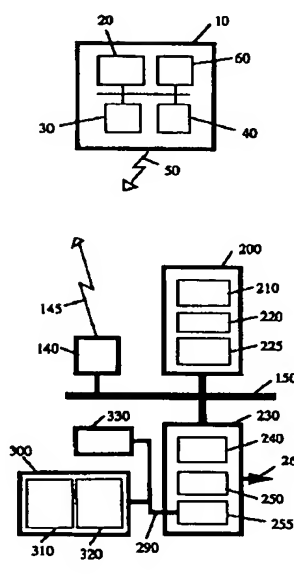
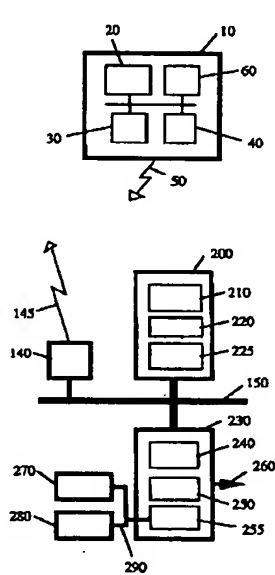
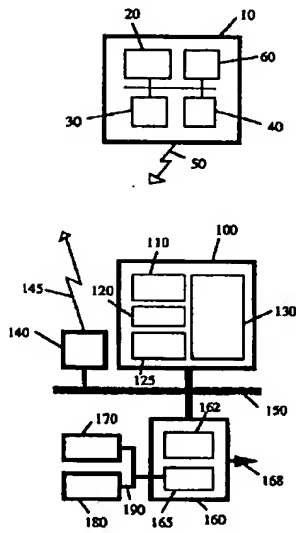
【図4】

図1

図2

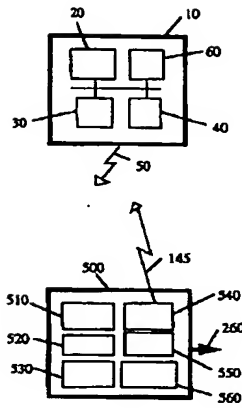
図3

図4



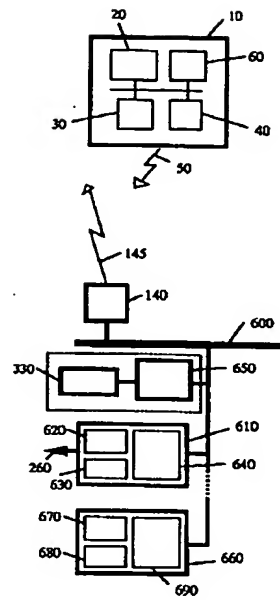
【図5】

図5



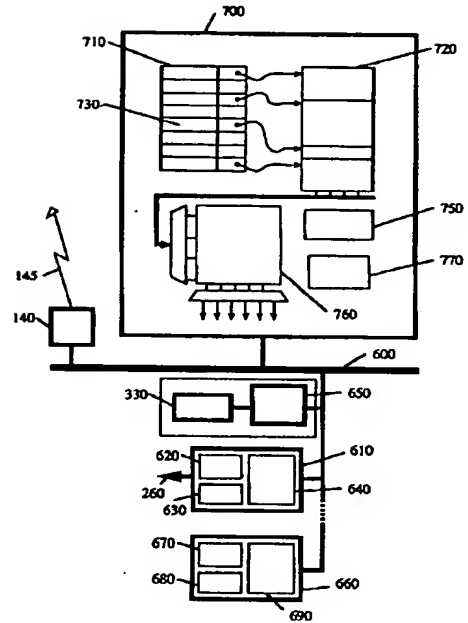
【図6】

図6



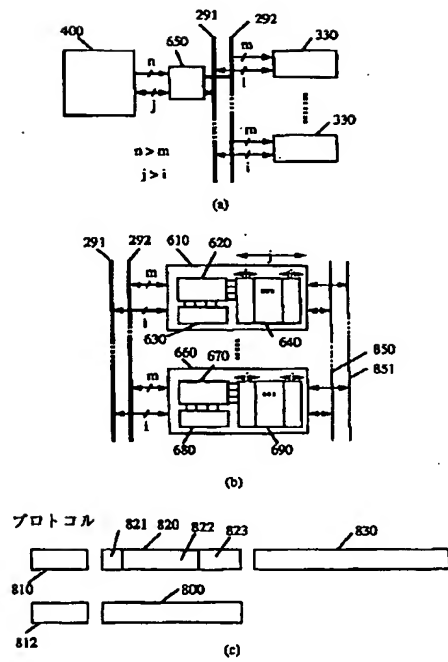
【図7】

図7



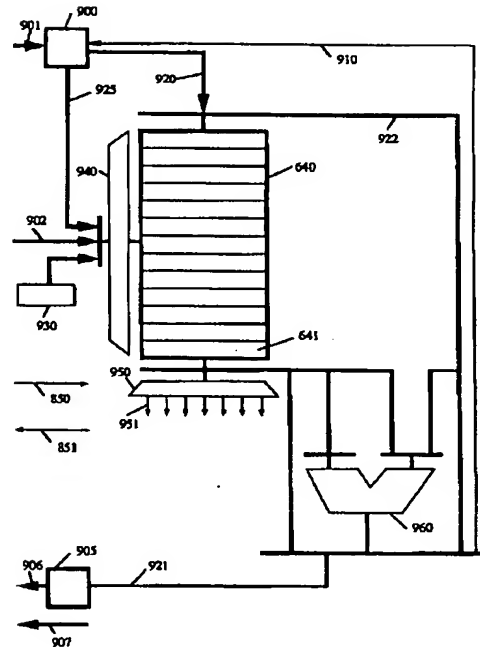
【図8】

図8



【図9】

図9





【図10】

図10

